

AMENDMENTS TO THE SPECIFICATION:

Page 12, amend the paragraph beginning at line 2 as follows:

This document is intended as a design specification for a software tool to be used by ARM, initially in the SoC group. In addition to specifying the product, it explains the benefits of the tool. This document specifies the requirements, but does not attempt to fully document the detail of the implementation.

Page 12, amend the paragraph beginning at line 7 as follows:

In a typical SoC ASIC or peripheral project, verification forms a significant part of the design effort. Once a test bench and a suite of simulations have been defined, it is likely that the same simulations will be run periodically as the design evolves. This approach, which is known as *regression testing*, verifies that modifications do not affect unrelated parts of the design, and also possibly that different views are equivalent (eg, netlist/RTL, best-case timing/worst-case timing). As the design nears completion, the pressure to complete regression testing as quickly as possible increases, and the number of tests to be performed increases. With each test typically taking many hours to run, any improvement in simulation speed ~~would~~will be advantageous.

Page 14, amend the paragraph beginning at line 3 as follows:

TBGen processes a POC (Print On Change) file from a simulation run and a file defining the inputs and sampling information for outputs. Bidirectional signals can be accommodated by tracing the bidir-enable signals. The POC file is then processed by TBGen to generate an output file which can be of 2 types, as required by the user:

- VHDL (Very High Speed Integrated Circuit Hardware Description Language) or ~~Verilog~~VERILOG® file (a hardware description file used to design electronic systems) containing stimuli & expected responses;
- Native simulator command file. (TBD: this format should deliver the ultimate in simulation speed but it may not be worthwhile implementing this format. This format is not considered any further in this document).

Page 16, amend the paragraph beginning at line 9 as follows:

Since this tool provides a portable method of delivering an obfuscated verification suite & a means of improving the throughput of regression testing, it will have value to anyone who designs or uses IP blocks: SoC's ASICs ASSPs, peripherals. ~~Since the tool only eases part of the design process, rather than being particularly key to the process of SoC design, this is a tool which ARM could market as a CAE product to the IP community.~~

Page 16, amend the paragraph beginning at line 17 as follows:

Considerable thought needs to be given to the features that TBGen will support to ensure that it will be both *useful* and *usable* for large designs and long simulations. The tool must be able to convert large POC files quickly and without excessive memory usage or the use of intermediate files (file i/o is unacceptably slow). The tool ~~must be~~is designed such that the output file does not consume excessive disk space. The size of this file is determined by the number of i/o, the length of the simulation, and the i/o activity.

Page 17, amend the paragraph beginning at line 1 as follows:

TBGen *must* be appropriate to large designs and long simulations. Tricks ~~need to be~~are used to minimise the size, for example: a/ Alias real i/o names to compressed names (eg I0, O1, etc). b/ Generate clocks in 1-line statements or simple models. c/ Only sample outputs (& bidirectionals when they are outputs) at the required times, and under the control of the TBGen user, ie provide flexibility here. An alternative to sampling outputs is checking for the transition times (discussed later). d/ Use Verilog/VHDL setup/hold checkers & 1 data-value check rather than multiple data value checks. e/ Group related signals together as buses to minimise the number of lines in the TBGen output file. A graphical front-end ~~should be considered~~could be used to ease the process of generating the TBGen control files. Some prototyping & experimentation will be necessary before the full technical specification for TBGen can be finalised.

Page 19, amend the paragraph beginning at line 5 as follows:

In addition to strobing output signals based on other signals, TBGen will provide a setup/hold feature whereby the value is additionally tested before and after the actual strobe instant. This accommodates variations in the strobe line timing, which should also move the setup and hold sampling points. This ~~could~~can be implemented by detecting the time at which a signal changes and comparing this with the expected time. Since there will typically be only a few different sets of setup and hold conditions for pins in a particular SoC or IP block, TBGen will allow the user to define timing specifications, and later associate these with particular pins.

Page 20, amend the paragraph beginning at line 14 as follows:

TBGen ~~should~~can support a feature whereby signals which are busses or which share the same strobe requirement can be grouped together to keep the TBGen output file compact. ~~Any other ideas to minimise the file size will be considered!~~

